NASA Technical Memorandum 86036

The Development of an Airborne Instrumentation Computer System for Flight Test

Glenn A. Bever

April 1984

NASA
National Aeronautics and
Space Administration

NASA Technical Memorandum 86036

The Development of an Airborne Instrumentation Computer System for Flight Test

Glenn A. Bever
NASA Ames Research Center, Dryden Flight Research Facility, Edwards, California 93523

1984

NASA
National Aeronautics and
Space Administration
Ames Research Center
Dryden Flight Research Facility
Edwards, California 93523

THE DEVELOPMENT OF AN AIRBORNE INSTRUMENTATION COMPUTER SYSTEM FOR FLIGHT TEST

Glenn A. Bever
NASA Ames Research Center
Dryden Flight Research Facility
P.O. Box 273
Edwards, California 93523
U.S.A.

SUMMARY

Instrumentation interfacing frequently requires the linking of intelligent systems together, as well as requiring the link itself to be intelligent. The airborne instrumentation computer system (AICS) was developed to address this requirement. Its small size, approximately 254 by 133 by 140 mm (10 by 5¼ by 5½ in), standard bus, and modular-board configuration give it the ability to solve instrumentation interfacing and computation problems without forcing a redesign of the entire unit. This system has been used on the F-15 aircraft digital electronic engine control (DEEC) and its follow-on engine model derivative (EMD) project, and in an OV-1C Mohawk aircraft stall-speed warning system. The AICS is presently undergoing configuration for use on an F-104 pace aircraft and on the advanced fighter technology integration (AFTI) F-111 aircraft.

NOMENCLATURE

AFTI	advanced fighter technology integration	LSB	least significant byte
AICS	airborne instrumentation computer system	LSI	large-scale integration
ARINC-429	Aeronautical Radio, Inc. specification number	MIL-STD-1553	military standard number
DEEC	digital electronic engine control	MSB	most significant byte
DMA	direct memory access	PCM	pulse code modulation
EEPROM	electrically erasable programmable read-only memory	PROM	programmable read-only memory
EMD	engine model derivative	RAM	random access memory
EPROM	erasable programmable read-only memory	RMDU	remote multiplexer digitizer unit
I/O	input/output	RS-232C	Electronics Industry Association standard number
LCD	liquid crystal display	TTL	transistor-transistor logic
		USART	universal synchronous/asynchronous receiver transmitter

1.0 INTRODUCTION

The increasing digital nature of aircraft being tested at NASA Ames Research Center's Dryden Flight Research Facility has increased the need for flexible digital instrumentation systems. Test aircraft typically have digital systems of unique design on board and researchers need to derive data from those systems. These aircraft often have minimal space for flight-test instrumentation. This paper addresses the need for a system that is capable of interfacing special digital systems to pulse code modulation (PCM) systems and provides onboard engineering calculations and display. The development of the system is discussed, and several examples are described.

2.0 THE PROBLEMS

Aircraft instrumentation systems used at NASA Ames Dryden are built either in-house or by a contractor. When a project is designed in-house, NASA has the choice of selecting those systems that correspond well to engineering experience and data reduction methods. When an instrumentation system is contractor built (off-site), it is frequently of a unique design, or of a design that does not lend itself to merge gracefully with the Ames Dryden data systems.

Because space on an aircraft is limited, there is typically very little room to house an instrumentation system.

Project requirements, regardless of where they originate, tend to be evolutionary. Because of unforeseen data requirements, new ideas, or lack of foresight, instrumentation systems may require modifications. What appears to be a minor modification can cause significant changes in the overall system.

Real-time data reduction is desired on a number of flight projects that are flown at Ames Dryden. The justification for these requirements include: saving flight time by immediately determining if a data run has produced the required data; helping the pilot identify a potential problem in his flight systems and

allowing him to take corrective action; dynamically changing the flight plan based on observed data; and reducing the postflight data reduction tasks.

Research flights are becoming longer in duration and are increasing in frequency. This puts real-time data reduction facilities at a premium. Postflight data reduction is a time-consuming process and requirements for use of this facility add significant delays in providing researchers with flight data. Methods to shorten the time requirements on these facilities are therefore desirable.

Finally, manpower and budget constraints are increasingly becoming a problem.

In short, a small, versatile, programmable, and inexpensive digital interface and calculation system is needed.

3.0 A SOLUTION

Solutions to problems are derived from experience, knowledge, and the use of available equipment. The solution presented here is not necessarily in the order of that which was followed, but all of the items were essential in the production of the final result. At all times, simplicity and flexibility of design were kept in mind, as was serviceability of the equipment. An effort was made to solve many instrumentation problems at once by developing multipurpose and standardized units.

3.1 Bus Selection

Several buses were considered. Design goals included simplicity of bus architecture, small size, and availability of low-cost commercial boards and prototype building cards. The STD bus (Ref. 1) met those design goals. It was designed to work with 8-bit microprocessors and has an 8-bit data path. An 8-bit microprocessor (as opposed to more complicated, newer 16-bit microprocessors) was deemed adequate for most flight test applications. The bus supplies circuit boards with +5 V, +12 V, and -12 V. No onboard regulators were required for transistor-transistor logic (TTL) circuitry, thus saving board space.

3.2 Enclosure

The computer enclosure has to be as small as possible to fit in the limited space of most aircraft, yet be large enough to accommodate three wire-wrapped prototype cards or six printed circuit cards. This requires a minimum of 15.875 mm (0.625 in) spacing between card slots. As prototype cards are debugged (problems located and fixed) and made into printed circuits, more space is available for additional prototypes. In this way, the same enclosure can be used by both prototype and final systems. A number of these boxes were manufactured at Ames Dryden without detailing the mounting of power supplies and connectors. This gives each project the flexibility to determine what interface connectors to use on the AICS. Signals can be functionally grouped in different connectors. Inexpensive off-the-shelf connectors can then be used, which reduces the problem of special connectors being out of stock.

3.3 Microcomputer Board Selection

A number of interface cards were commercially available for the STD bus, but the available microprocessor cards were not functional in a very small system. Since as much room as possible should remain in the enclosure for project-dependent interfaces, the basic general purpose computer and standard input-output circuitry should be contained on one card. Because some projects require the processing of significant amounts of real-time complex engineering equations, an arithmetic or floating-point processor was used. Since the card needed to withstand rugged flight environments, and one was not available that satisfied all of the above requirements, a decision was made for an in-house design of the microcomputer board.

4.0 HARDWARE SYSTEM DESCRIPTION

The AICS consists of an enclosure approximately 254 by 133 by 140 mm (10 by 5¼ by 5½ in) and contains a single-board microcomputer that plugs into an industry-standard STD backplane. This backplane accommodates either commercially available or specially designed interface boards. The enclosure itself can be configured to fit the needs of the individual project with respect to power supplies and external connections. Typical project configurations yield an AICS power consumption of under 50 W. Having the power supply determined by the project allows state-of-the-art power converters to be used as they become available. Fig. 1 is an internal view of the F-15 digital electronic engine control (DEEC) prototype AICS and Fig. 2 shows an enclosed AICS as used in the OV-1C aircraft.

The heart of the AICS system is a microcomputer board that plugs into a wide-spaced STD bus backplane that contains six card slots. There are enough standard peripheral components on the processor board so that the five remaining STD bus slots can be devoted to special system requirements.

4.1 Microcomputer Board Description

The single-board computer uses six-layer printed circuit technology to accommodate a number of large-scale integration (LSI) components on a single board measuring 114.3 by 165.1 by 1.6 mm (4.5 by 6.5 by 1/16 in). The board contains:

1. One 8085A microprocessor.

2. One 8231A floating-point arithmetic processor.

3. 8 kilobytes of 2732 erasable programmable read-only memory (EPROM).

4. 2 kilobytes of either 2716 EPROM or 6116 random access memory (RAM) plus 256 bytes of RAM in an 8155 RAM input-output (I/O) port.

5. Two 8251 universal synchronous/asynchronous receiver transmitters (USART) that can be used either with TTL signals or RS-232C level signals provided by the 75188 line driver and the 75189 line receiver.

6. Three 16-bit programmable timers in an 8253 programmable timer (two used for USART bit rate control).

7. 22 bits of parallel I/O using the 8155.

8. Address decoding using 82S131 programmable read-only memory (PROM).

9. STD bus interface.

10. Flight-qualified, low-profile sockets that every integrated circuit plugs into.

11. Board configuration controlled by jumpers on the user interface connector.

Fig. 3 is a simplified block diagram of the single-board computer and Fig. 4 is a photograph of the multilayer board. One reason the 8085A microprocessor (Ref. 2) was chosen was because of its superior interrupt circuitry. It allows direct inputs for one unmasked and three masked interrupts with various combinations of level and edge triggering, as well as allowing one line for up to seven externally vectored interrupts. Because instrumentation projects are intensely real-time oriented, interrupts are used extensively. Good priority interrupt circuitry on the microprocessor chip can reduce the microcomputer board component count.

The 8085A is clocked by a crystal operating at a frequency of 6.144 MHz. It yields a microcycle time of 326 ns. A typical instruction completes in 2 to 3 μs. Address, data, and control lines (such as Read, Write, Clock, I/O-Memory select, Address Latch Enable, and Reset) are bused to all of the peripheral circuits.

The 8155 provides two 8-bit and one 6-bit parallel input/output ports, 256 bytes of static RAM, and a 14-bit programmable timer for the board (Ref. 2). Two 2732 EPROMs (Ref. 2) provide the board with 8 kilobytes of program memory, with the remaining 2 kilobytes of memory being either a 2716 EPROM (Ref. 2) or a 6116 RAM (Ref. 3).

Two 8251 integrated circuits (Ref. 2) provide USARTs to the board. Their bit rates are determined by the 8253 programmable timer which allows separate control of the two 8251 bit rates. The USARTs can be programmed to operate in either a synchronous mode at 0 to 64,000 bits per second or in an asynchronous mode at 0 to 19,200 bits per second. The 75189 and 75188 integrated circuits (Ref. 4) provide RS-232C I/O signal level interfacing.

The onboard address decoding is provided by three 82S131 PROMs (512 by 4 bits each). Use of PROM decoders allows maximum flexibility in determining onboard addresses (there is no fragmentation of memory or I/O addresses — all addresses are contiguous and unique) as well as allowing offboard direct memory access devices to access the computer board's memory. Use of discrete logic to accomplish these tasks would take up too much room on the board.

Probably the most important factor in making this board universally applicable to many real-time instrumentation systems is the use of the 8231A arithmetic processor. The 8231A integrated circuit is a powerful processor that will perform single- and double-precision fixed-point, as well as floating-point arithmetic and scientific calculations (Ref. 2). Thus all of the mathematical operations can be offloaded to the arithmetic processor which contains a 32-bit wide internal data path. Benchmark testing has indicated that a ten- to twentyfold-savings in execution time can be realized by using the arithmetic processor (Ref. 5). Execution times of floating-point instructions in the arithmetic processor vary considerably; for example, the floating-point multiply instruction takes approximately 50 μs and a cosine operation takes 1.34 ms (Ref. 2).

The 8231A is interfaced to the 8085A through the 8-bit data bus, control lines for Read and Write, Clock, Chip select, and the least significant address line. This address line, in conjunction with Read and Write signals, tells the 8231A what types of operations it will be doing; (for example, data entry or command). The 8085A passes data and commands to the 8231A, and receives results back from the 8231A. Data is stored internally on a stack in the 8231A. The data stack is either 16 bits or 32 bits wide, depending on the operation. The 8085A is typically programmed to wait for the completion of each command it sends to the 8231A. This is the most straightforward way to program it. The 8085A could, however, be programmed to initiate the 8231A instruction, continue with 8085A instructions, and check the status of the 8231A only when its result is required by the 8085A. This would increase the information processing rate (throughput) in some cases, but generally the arithmetic operations are done contiguously. Therefore, the program throughput would not be enhanced by this alternate scheme. It would, however, require a larger code space because of the extra status checks required.

4.2 Telemetry Interface

Telemetry systems at Ames Dryden use data words that are between 8 and 12 bits long — typically, 10 bits. Onboard computer systems that need to interface to the telemetry system frequently have word lengths that are 16 bits long. Two asynchronous activities are taking place in the system: data coming into the AICS from the onboard computer or data system, and data being passed from the AICS to the

telemetry system. Because these two activities are asynchronous, it is necessary to ensure that a data word of 16 bits does not get half transferred in (8 bits) at a time when the telemetry system is requesting a whole transfer out. This is accomplished by latching all 16 bits and transferring them all at the same time to a 16-bit memory. When the telemetry system requests data, all 16 bits are read out of memory at once and stored in a register where the telemetry system can access them.

To interface the AICS with a remote multiplexer digitizer unit (RMDU) PCM system, two boards were designed. Fig. 5 is a simplified block diagram of the circuitry on these boards. One board contains the memory accessible by the PCM system and the other contains the direct memory access (DMA) controller that the PCM system uses to interrogate the PCM memory. The basic operation can be summarized as follows: The central processor places data in the two 8-bit latches, then simultaneously writes them into the PCM accessible memory as a single 16-bit value. The PCM input port is organized in 10-bit units. When the PCM system requests a value through a PCM word request signal, the "glitch" detector (a glitch is a spurious asynchronous signal that is to be ignored) verifies the validity of the request. A divide by two latch activates the DMA controller every second PCM word request. The DMA controller presents the requested 16-bit data to the PCM input port within 4 μs. The most significant 10 bits are taken by the PCM on that request. The next PCM word request multiplexes the lower order 6 bits without requiring another memory read. The PCM end of cycle signal resets the divide by two latch to ensure synchronization of the DMA activity with the telemetry system. It also terminates the DMA activity for that cycle. This signal is required only once to synchronize the telemetry data map with the interface, but continuous use of this signal ensures that transient improper operation will not cause the interface to lose synchronization for very long.

4.3 Voice Output

This board was designed to interface the processor with a speech synthesizer through the STD bus. Fig. 6 is a simplified block diagram of this board. It uses a Votrax SC-01 phoneme speech synthesizer (Ref. 6) that has 64 different phonemes. Circuitry has been added to allow extended pitch control. An 8755 integrated circuit (Ref. 2) is used to control various onboard functions, as well as provide 2 kilobytes of onboard EPROM for word tables or additional programming space. Use of a phoneme-based speech synthesizer allows customized vocabularies to be programmed and tested inexpensively. (Phoneme synthesizers sound mechanical, yet are intelligible.)

5.0 Software Design

Programming the AICS for a specific project is a major part of the application effort. To effectively accomplish that, software development tools are required. One of the tools available at Ames Dryden is a powerful relocating cross assembler on a minicomputer. Instead of filling up the AICS memory with an existing operating system and spending a lot of time working around it (and locating the problems, or "bugs"), the system software was completely designed and tailored to the particular needs of flight test instrumentation. This optimized the code size, as well as providing in-house knowledge of the inner software workings.

The instrumentation requirements that dictate the software design include: functionally optimized code (low overhead); the ability to stand alone (nonvolatile program memory); optimal utilization of hardware; and simplicity. In order to achieve functionally optimized code, all code was written in assembly language. High-level computer language compilers that optimize code were not available for this system. Extensive use of user-definable operation codes (macros) in the macro assembler raised the level of programming to almost that of a high-level language, but retained optimizing characteristics by assembling only what was needed. The use of a librarian allowed linkage of only those routines required by the programming. The use of a microprocessor hardware emulator allowed program development to proceed without requiring the program development code to be part of the microprocessor software. The use of the real-time hardware emulator is an indispensable tool in developing microprocessor systems in a time-effective manner.

Reasons for programming in a high-level language (such as FORTRAN or PASCAL) include the need for: formatted input/output; floating-point formula calculations; and ease of programming and debugging (finding errors). Using macro coding, formatted input/output and floating-point formula calculations were accomplished. Ease of programming usually implies separation from hardware, which, by the nature of this system, is undesirable. A reasonable compromise has been achieved by the use of libraries of macros and subroutines.

6.0 EEPROM DATA STORAGE TECHNIQUE

In data acquisition systems requiring small amounts of data, electrically erasable programmable read-only memory (EEPROM) is useful in providing nonvolatile storage. It allows flight-dependent documentation data and calibration coefficients to be entered without requiring removal of components for programming. It does not require battery backup, it is inexpensive, and because it is solid state it has no moving parts. The technique of using EEPROM for data storage was used on the OV-1C aircraft stall warning system to store flight parameters at moments determined by the pilot. After engine shutdown, data could be off-loaded (in both uncalibrated and calibrated forms) directly from the AICS to a printer to provide hardcopy of the flight data.

7.0 ENVIRONMENTAL CONSIDERATIONS

Exhaustive environmental tests have not yet been conducted on the AICS. However, one configuration presented in section 8.2 has been vibrated to NASA process specification 21-2, curve A (±2g from 32 to 500 Hz). Forced air ventilation of at least 0.850 m^3/min (30 ft^3/min) has been used on all flying systems. All components used on the microprocessor board are available in military-specified (MIL-SPEC, harsh environment) form. The temperature regimes of current applications have allowed the use of commercial grade components, thus saving development cost.

8.0 APPLICATION EXAMPLES

8.1 F-15 EMD (DEEC)

The AICS is presently being used on the F-15 aircraft DEEC and its follow-on engine model derivative (EMD) project. This aircraft is fitted with two DEEC computers. Fig. 9 is a simplified block diagram of the F-15 DEEC instrumentation system. The AICS is acting as the interface between the two DEEC computers and the PCM system. Each of the DEEC computers sends 16-bit data asynchronously to the AICS in 8-bit quantums at 9600 bits/sec. Each data frame consists of 100 16-bit words, including 1 16-bit synchronization word. The USARTs on the AICS microprocessor board receive this data and interrupt the microprocessor, which in turn decommutates the data and places it in the memory on the PCM memory card. The telemetry system retrieves this data as described in section 4.2. Status words are formed by the AICS to determine the health of the system, including program status, data-stream status, and synchronization status. This status information is passed to the telemetry system for real-time evaluation.

The number of words per frame of asynchronous data, as well as the number of DEEC engines (one or two) tested, has varied. Since the AICS is a microprocessor-based design, only software modifications have been required to reconfigure it for the different configurations.

8.2 OV-1C Aircraft Stall Warning

The AICS is also being used to test a stall warning concept aboard a United States Army OV-1C Mohawk aircraft (Ref. 7) in a joint NASA/Army project. Fig. 8 is an overall block diagram of the OV-1C stall warning system. This system represents the most complex use of the AICS to date, and serves to illustrate the potential power of the AICS. It provides nearly all of the functions associated with real-time data collection and display including:

1. data acquisition via two 10-bit digital inputs;
2. data acquisition via nine analog inputs;
3. sensor calibration assistance;
4. real-time engineering unit calculations;
5. real-time control of synchro-driven cockpit indicator needles displaying airspeed and stall speed;
6. a voice synthesized aural warning of approaching stall to the pilot;
7. audible sensor limit warning to the flight test engineer;
8. nonvolatile data storage;
9. real-time data display of uncalibrated instrumentation counts, as well as engineering unit data; and
10. postflight dump of the raw data or engineering unit data for immediate analysis.

Use of a hand-held liquid crystal display (LCD) keyboard data entry and display unit (Fig. 9) allows the flight test engineer to access flight data and enter selected coefficients.

8.3 Advanced Fighter Technology Integration (AFTI) F-111 Aircraft

The AFTI program is studying the concept of a variable camber wing on an F-111 airplane. It has a complex control system that is controlled by two airborne computers. This system was designed by a contractor and is another example of the requirement of a unique interface to NASA's PCM system. The AICS hardware used for this interface task is nearly identical to the set used on the F-15 aircraft. It was only necessary to add one line receiver card. Because of the flexibility inherent in the microprocessor-based design of the AICS, the remaining modifications required by the project were completed in the software.

9.0 PLANNED APPLICATIONS

9.1 F-104 Pace Aircraft

Development is underway to use the AICS as a real-time calculation and display system for aircraft pacing work. It would, based on pitot/static pressure sensor input, provide altitude, airspeed, and position-error corrected Mach number display to the pilot or flight test engineer, as well as optionally insert this information into the PCM system for comparison with ground data calculations. Fig. 10 is a simplified block diagram of this instrumentation system.

9.2 Integrated Sensor System Front Panel Controller

The microcomputer board developed for AICS is also being used in a ground-based project as a controller for data entry and display via an ARINC-429 data bus (Ref. 8). The ARINC-429 bus is used as the communication bus for state-of-the-art strapdown inertial navigation systems being used at Ames Dryden. The same software development tools that were used for the airborne AICS are also used for this project.

10.0 PLANNED ENHANCEMENTS

Planned enhancements to the system include the ability to: interface with different types of PCM systems (in addition to the RMDU); interface with the MIL-STD-1553 data bus (Ref. 9); and decommutate data from onboard PCM systems so that the AICS can take advantage of telemetered data rather than requiring the AICS to interface to the sensors directly.

Time-integrated parameters, such as current aircraft weight, could be calculated in real time on an aircraft using the AICS. These parameters could be passed on to the flight recorder or ground stations. This would simplify postflight data reduction by allowing time-integrated flight parameters to be observed without requiring that a flight tape be replayed for integration.

To increase the system information processing rate, multiprocessing could be used in the AICS. Although the STD bus, by virtue of its simplicity, is not designed for multiprocessing, it could be achieved by electrically splitting the STD backplane into two or more individual STD buses. The microcomputers on each bus could then be responsible for different processes and communicate with each other through their parallel or serial I/O lines on the user interface connector. Thus a division of labor would be achieved without altering the bus architecture or requiring substantial rewriting of system software.

Another approach for increasing the system's information processing rate is to use peripheral processors on boards that are interfaced to the host microcomputer board through the STD bus. The host microcomputer would act as overall system controller, but the peripheral processors could handle special purpose I/O requirements.

As LSI manufacturing techniques improve, more powerful microcomputer components are being built in smaller packages that use less power than previously possible. This offers the possibility of using these more powerful microcomputers in a system the size of the AICS. In order to maintain the current STD bus interface for compatibility, an 8-bit data path would need to be maintained off the microcomputer board, but a 16-bit (or larger) data path could be used on the microcomputer board itself to increase bandwidth.

11.0 CONCLUSIONS

Flexible aircraft instrumentation interfaces such as the airborne instrumentation computer system (AICS) are obtained by using a microprocessor-based system design. Large-scale integration (LSI) technology allows these computer systems to be small enough to fit in most test aircraft. Because of software tools, project modifications can frequently be made without requiring hardware redesigns. A common set of hardware can be used for several projects, thus decreasing the overall cost of engineering each system. The use of a floating-point arithmetic processor allows real-time engineering equation calculations to take place onboard the aircraft for display in the cockpit or for telemetering to the ground. This shortens the time requirements for a ground-based real-time data reduction facility. Onboard calculation of time-integrated parameters (such as gross weight) simplifies postflight data reduction by allowing time-integrated flight parameters to be observed without having to replay an entire flight tape to integrate them. This saves in postflight data reduction time. Significant savings in data analysis and instrumentation modification times are realized using this design.

REFERENCES

1. STD Bus Technical Manual and Product Catalog, 1982, Pro-Log Corporation, Monterey, Calif., pp. 1-1 to 2-5.

2. Component Data Catalog, 1982, Intel Corporation, Santa Clara, Calif.

3. IC Memories, 1980, Hitachi, San Jose, Calif., pp. 71-73.

4. The Linear and Interface Circuits Data Book for Design Engineers, 1973, Texas Instruments Incorporated, First Ed., pp. 8-95 to 8-102.

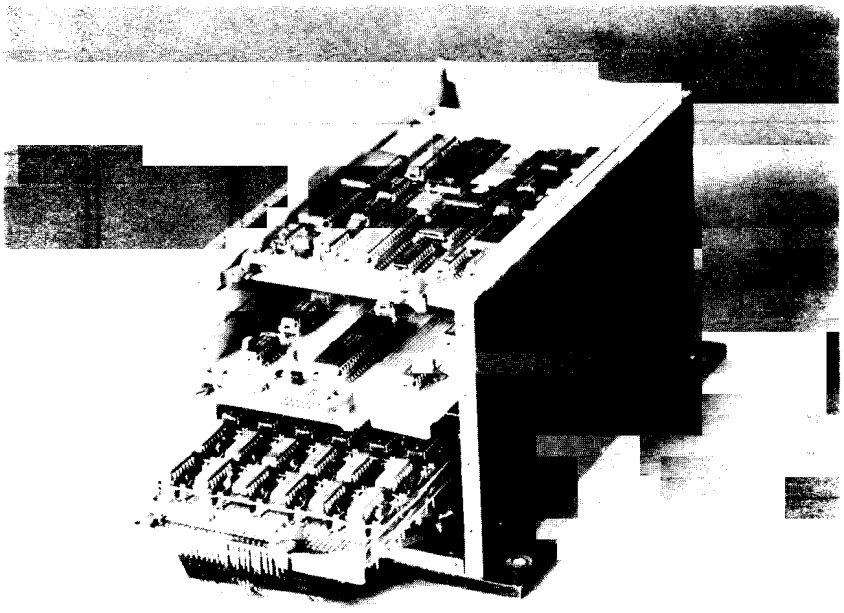
5. SC-01 Speech Synthesizer Data Sheet, 1980, Votrax, Troy, Mich.

6. Wilner, Douglass O. and Bever, Glenn A., An Automated Stall-Speed Warning System, 1984, NASA TM-84917.

7. Am9511A/Am9512 Floating-Point Processor Manual, 1981, Advanced Micro Devices, Sunnyvale, Calif., pp. 43 to 49.

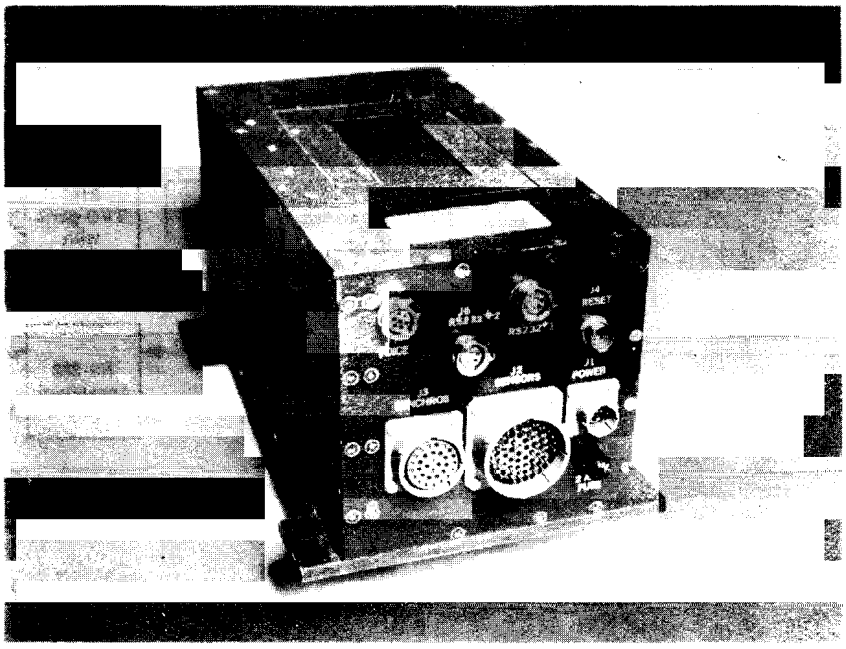
8. Mark 33 Digital Information Transfer System (DITS) ARINC Specification 429-6, 1982, Aeronautical Radio, Inc., Annapolis, Md.

9. MIL-STD-1553 Multiplex Applications Handbook, 1982, SCI Systems, Inc., Huntsville, Ala.



ECN 22636

Figure 1. AICS box (DEEC prototype configuration).



ECN 24940A

Figure 2. AICS box (OV-1C configuration).

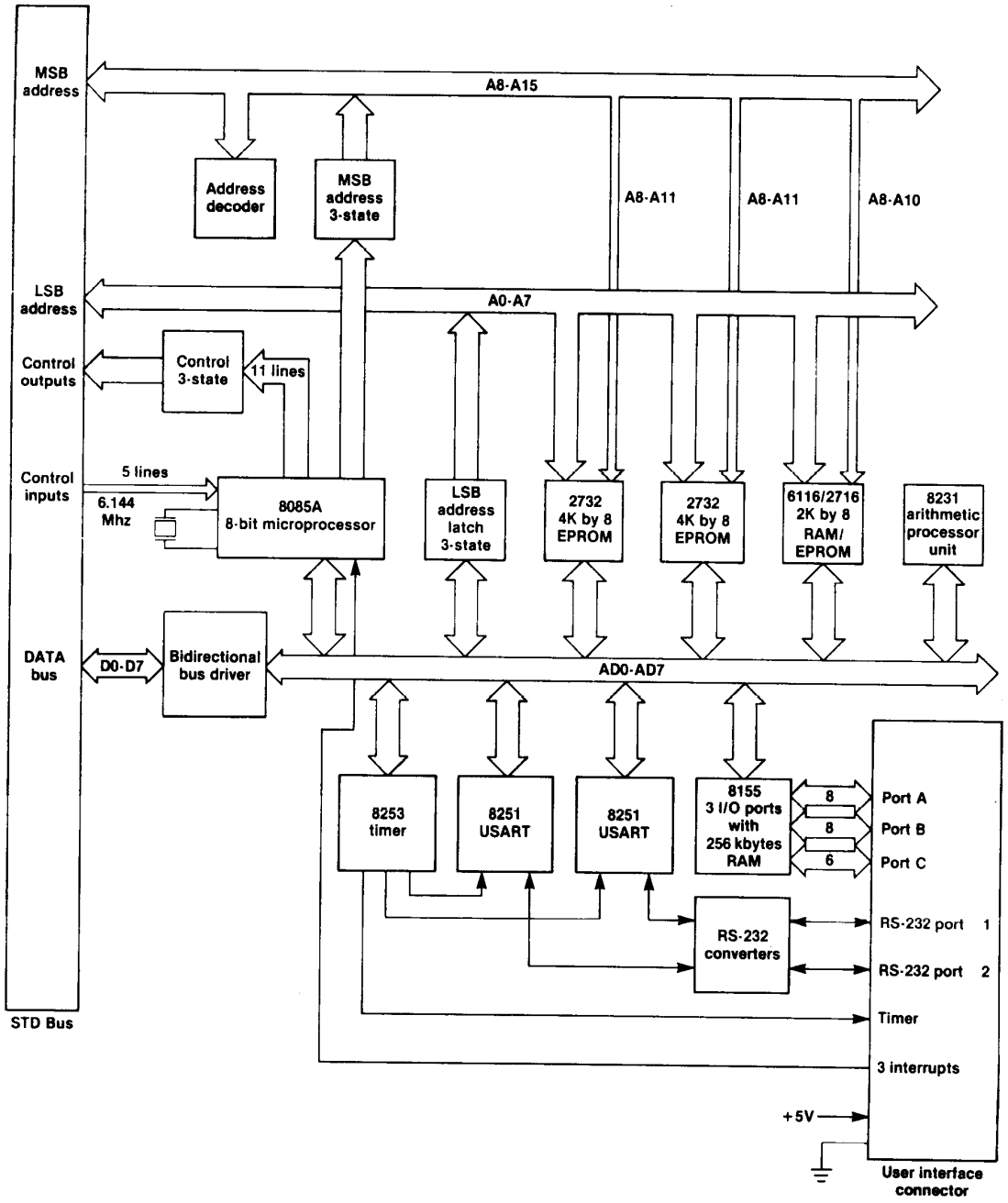
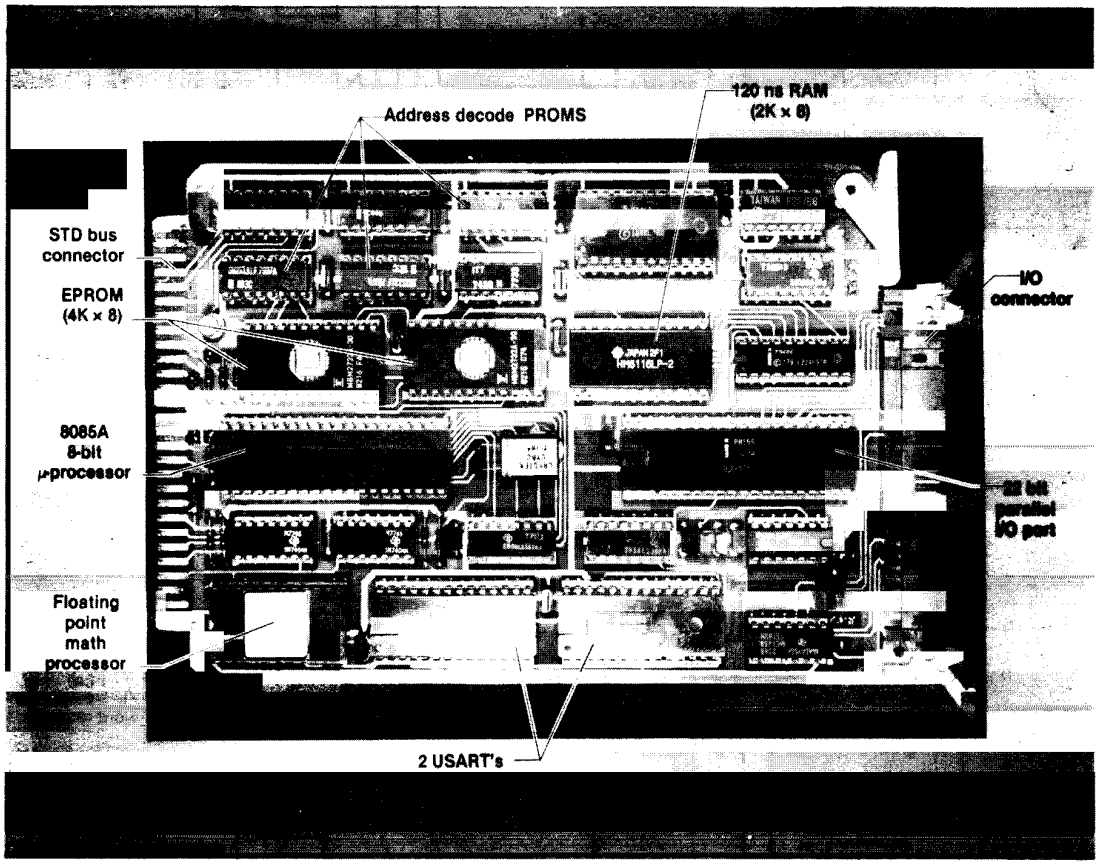


Figure 3. Simplified block diagram of a single-board microcomputer.



DFRF83-1432a

Figure 4. Single-board microcomputer.

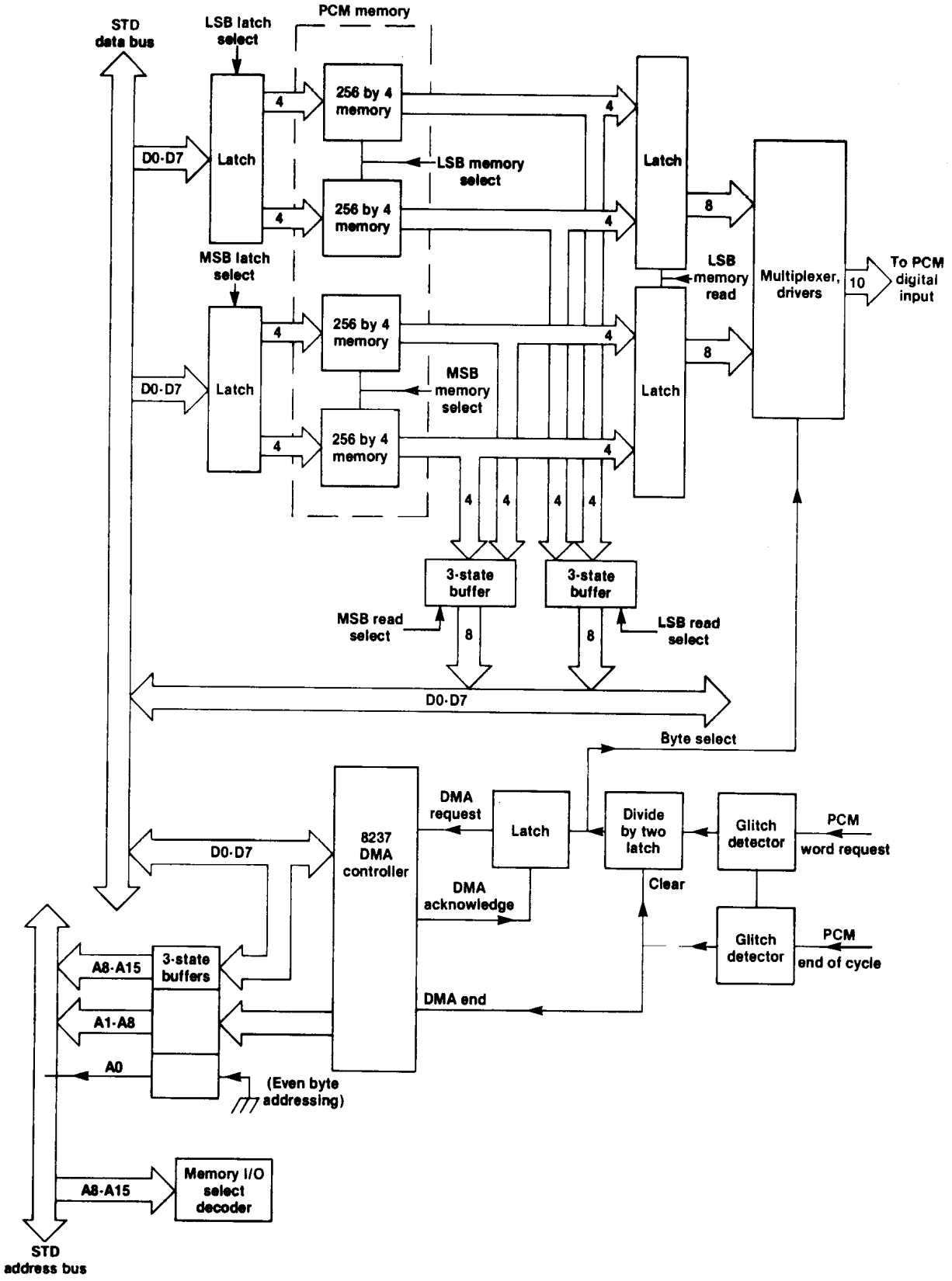


Figure 5. Simplified block diagram of a PCM interface.

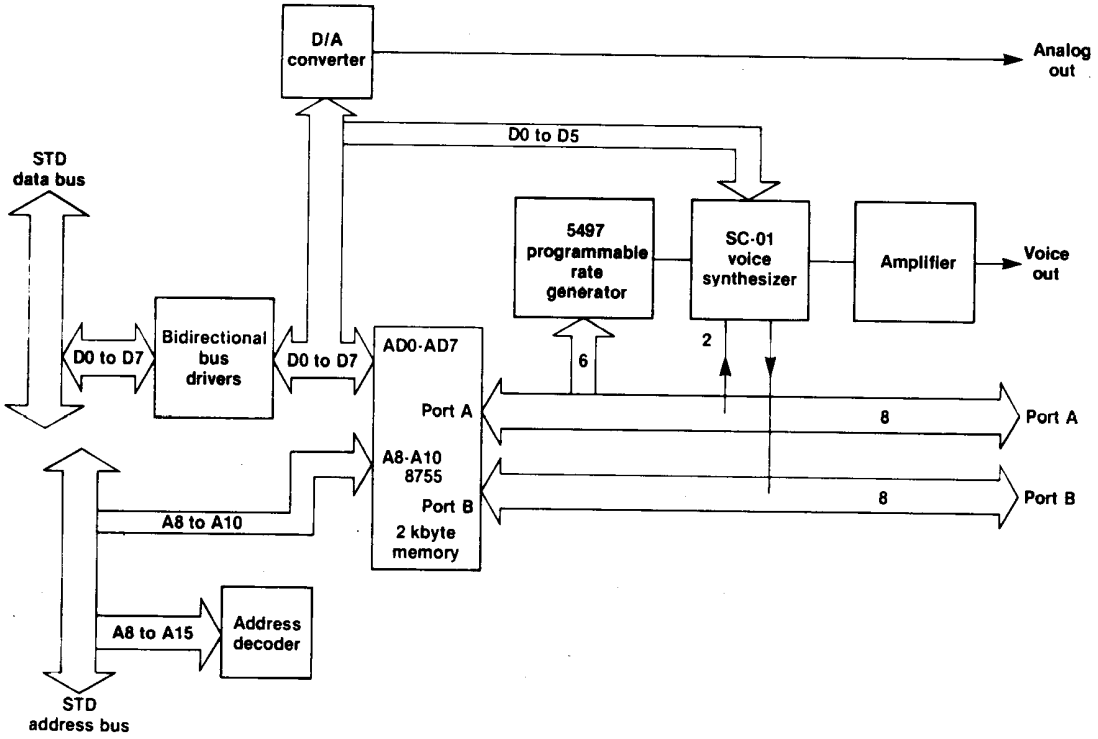


Figure 6. Simplified block diagram of a voice synthesizer.

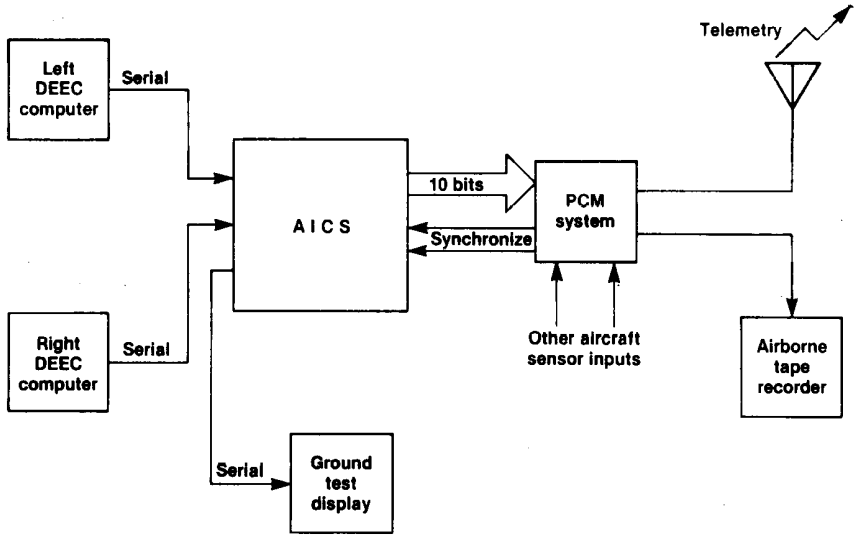


Figure 7. Simplified block diagram of the F-15 DEEC instrumentation system.

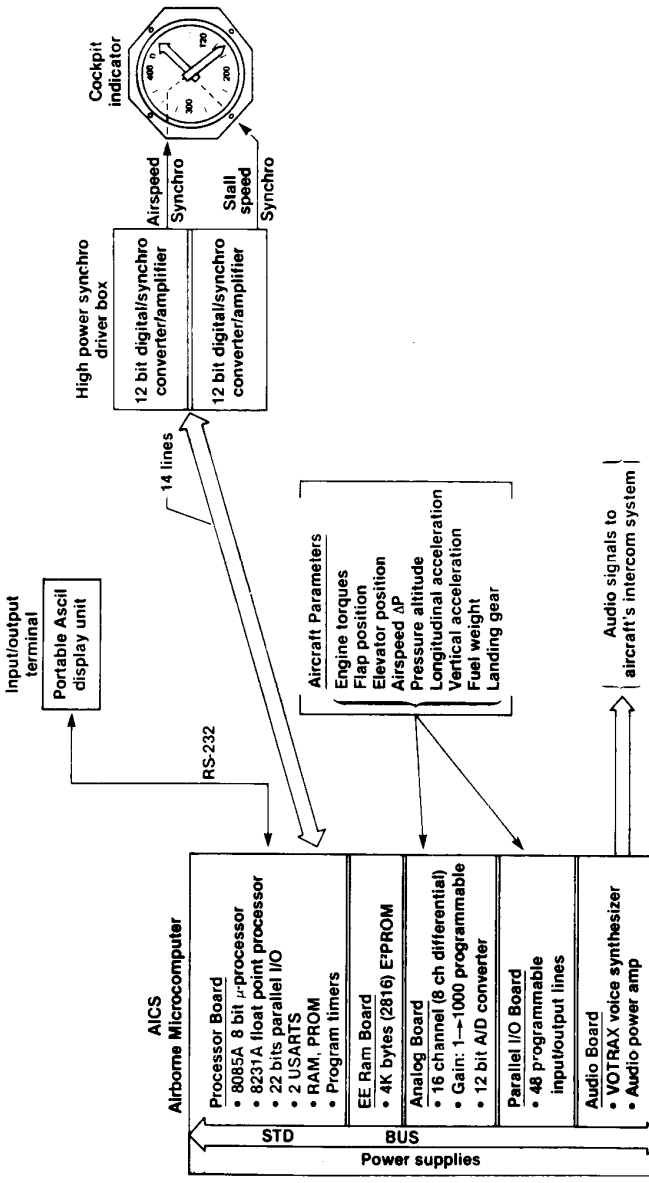
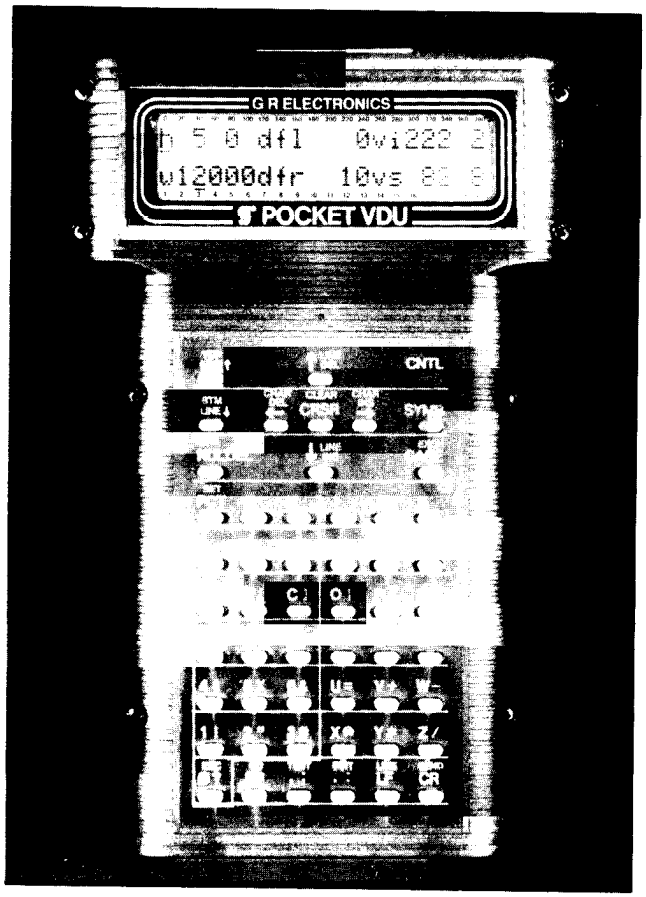


Figure 8. Overall block diagram of the OV-1C stall-warning system.



ECN 24926a

Figure 9. Portable LCD display unit.

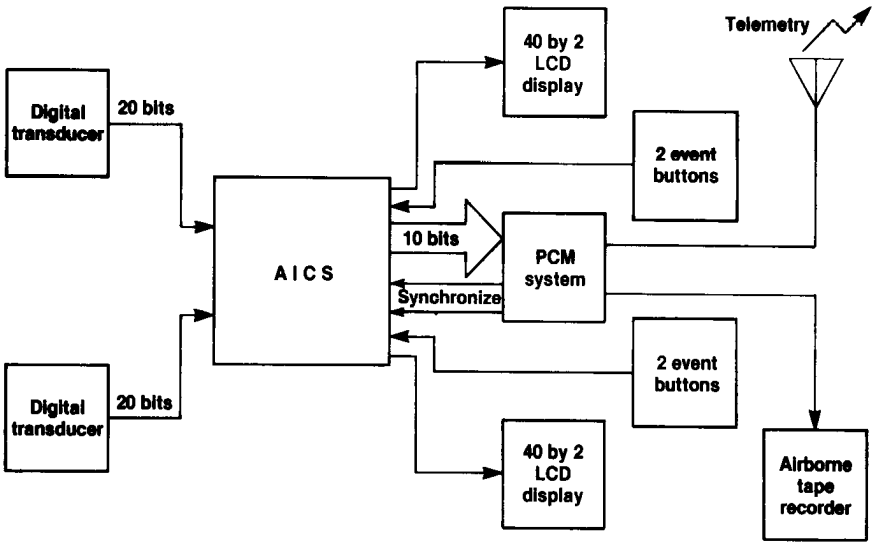


Figure 10. Simplified block diagram of an F-104 Pace instrumentation system.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-86036		

4. Title and Subtitle	5. Report Date
The Development of an Airborne Instrumentation Computer System for Flight Test	April 1984
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Glenn A. Bever	H-1233
	10. Work Unit No.
9. Performing Organization Name and Address	11. Contract or Grant No.
NASA Ames Research Center Dryden Flight Research Facility P.O. Box 273 Edwards, CA 92523	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, D.C. 20523	14. Sponsoring Agency Code RTOP 505-31-51

15. Supplementary Notes

This paper was prepared for the AGARD Flight Mechanics Panel Symposium, "Flight Test Techniques," Lisbon, Portugal, April 2-5, 1984.

16. Abstract

Instrumentation interfacing frequently requires the linking of intelligent systems together, as well as requiring the link itself to be intelligent. The airborne instrumentation computer system (AICS) was developed to address this requirement. Its small size, approximately 254 by 133 by 140 mm (10 by 5¼ by 5½ in), standard bus, and modular-board configuration give it the ability to solve instrumentation interfacing and computation problems without forcing a redesign of the entire unit. This system has been used on the F-15 aircraft digital electronic engine control (DEEC) and its follow-on engine model derivative (EMD) project and in an OV-1C Mohawk aircraft stall-speed warning system. The AICS is presently undergoing configuration for use on an F-104 pace aircraft and on the advanced fighter technology integration (AFTI) F-111 aircraft.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Microprocessor Arithmetic processing unit Flight instrumentation Voice synthesizer	Unclassified-Unlimited STAR category 06

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	14	A02

For sale by the National Technical Information Service, Springfield, Virginia 22161.